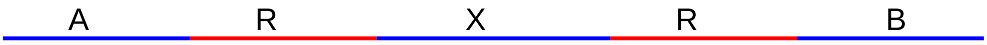
UNIVERSITY OF MARYLAND, COLLEGE PARK
ALEKSEY V ZIMIN, MICHAEL ROBERTS AND JAMES A. YORKE

ASSEMBLY RECONCILIATION METHOD

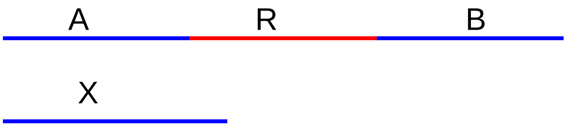
CE statistic

The CE (Compression/Expansion) statistic allows to detect misassemblies of compression, where a chunk of sequence is missing from the genome, or expansion, where there is an insertion of a chunk of sequence that does not belong there. Our assembly reconciliation techniques use CE statistic to detect misassemblies so that one could use other assemblies to correct them.

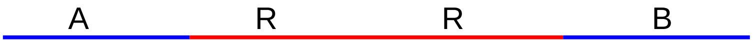
The great majority of misassemblies and gaps are the result of repeat regions. For example, assembly programs sometimes get confused by regions that look like this:



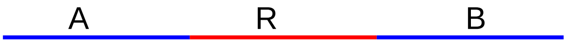
where the two R regions are nearly identical. If the assembly program cannot resolve this region it is likely to put a gap between A and B, possibly omitting both copies of R and X, or to create a "compression" misassembly:



where X ends up to be an unplaced contig or single-contig scaffold. Gaps are not misassemblies while compressions are. Tandem repeats of the form:



can also result in compression misassemblies, where the assembly program reports



Of course there can be many consecutive copies of R.

Collapsed repeats lead to missing DNA (either missing tandem or excision of unique region). The mate pairs can help us detect these regions. Based on where the two reads of an insert are placed, they pair is called *unhappy* if the number of bases between the mates differs from the expected by about 3 standard deviations. While assembly programs examine clusters of unhappy mate pairs for evidence of misassemblies, we have developed a statistical measure that enables us to find smaller misassemblies, and avoid false positive caused by mates at the boundaries of the normal distribution. As mentioned in Section B, we call this technique

Compression/Expansion (CE) statistic. Our CE statistic detects compression or expansion misassemblies using a given library of inserts. Our results have been distributed to the members of FAWG.

Definition. We select a library of inserts and, when both reads of the insert lie in the same contig, we compute the insert's (implied) length, based on where its reads are in the contig. The *global mean* is the average of the implied lengths of the inserts in long contigs, weighted in proportion to their *implied length*. We choose to use that weighting because longer inserts affect the count for more points. We also compute a standard deviation. We ignore all inserts whose length differs from the global mean by at least 6 standard deviations.

Here we treat the actual lengths of the inserts as if they are independent random variables and are independent of where they come from in the genome. Our investigations justify this approach. Our statistic is essentially the law stating that the variance of the sum of independent random variables is equal to the sum of their variances. We look at the inserts of a given library that span a given base in the genome. Using the read placement coordinates from the assembly, we compute the sample mean, i.e. the mean of the implied insert lengths, and we compute the "sample standard deviation of the mean", i.e., if *N* = the number of inserts in a sample, then

sample standard dev. = the global standard deviation / sqrt*(N)*

We compute our CE Statistic *C*,

C = (sample mean - global library mean) / sample standard dev

at each insert position in the assembly. *C* is the number of sample standard deviations by which the sample mean differs from the global mean. For *D. virilis*, one library of inserts had a mean length of about 37Kb with standard deviation of 3750 bases, and the genome had an average coverage of about 49 inserts, so *N* was often about 49. In such cases the sample standard deviation was sqrt(49) = 7 times smaller than the global standard deviation. The statistic was in effect 7 times as sensitive as looking at individual unhappy inserts.

At collapsed regions the statistic should be negative, while in regions where extra sequence has been inserted the statistic should be positive. We are therefore interested in the events where the value of *C* lies outside of some (pre-specified) interval about 0.

We determine and count the locations in the assembly where these events occur. Of course, we don't want to count nearby events as separate locations, since many of the same inserts are used when calculating the statistic. To make sure that the same inserts don't cause us to over count the number of distinct compressions or expansions we avoid counting pairs of events that occur too closely spaced. In any region of the contig whose length is the average (global) insert length, we count at most one event.

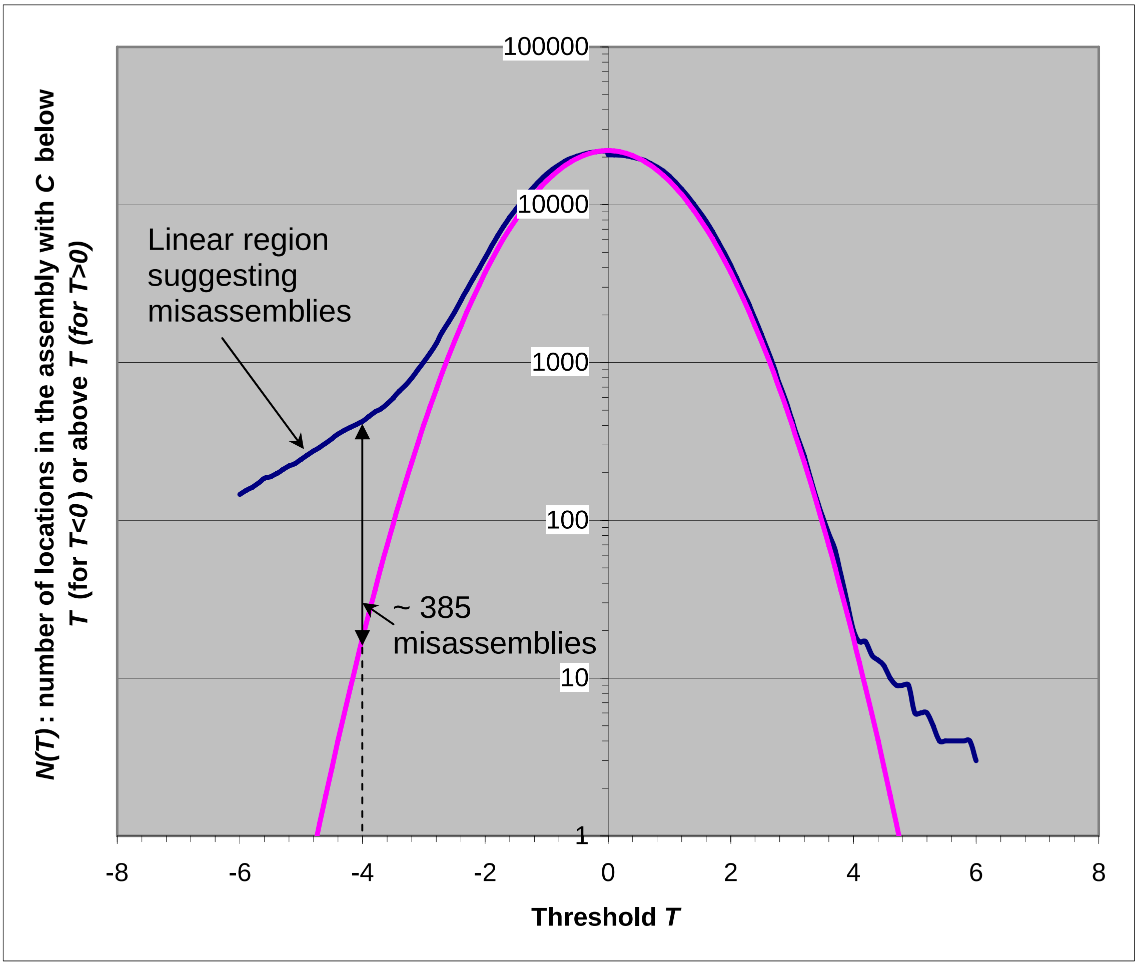


Figure 1. Distribution of the number of locations where |*C*| > |*T*| for *D. virilis* assembly created by using UMD Overlapper with Celera Assembler, for the library of 3500kb inserts.

Now there is a question: what threshold should we choose so that we detect mostly misassemblies with the CE statistic. Since insert sizes are distributed approximately normally, the local means are also approximately normally distributed. Hence in the perfect assembly the number of locations *N(T)* for which the CE statistic is above *T* or below -*T* must correspond to what would occur for a normal distribution. Misassemblies result in large tails (See Figure 1). In Figure 1 we plot the number of locations *N* as function of the threshold *T* for one library of inserts (3500's) for the draft assembly of *D. virilis* (blue curve) produced by our group in collaboration with VI and TIGR. For *T>0* we count all events in which *C* (value of the statistic above) >*T*, and for *T<0* we count all events in which *C<T*. We also plot a Gaussian fit (magenta) to the experimental (blue) curve. Since we chose the logarithmic vertical scale, we expect to see a good fit to a parabola (magenta) for all values of the threshold *T* that correspond to ordinary deviations within the draft assembly that are due to the distribution of the insert sizes in

the library. We expect the curve to become nearly linear for the values of T where we start seeing misassemblies. The approximate number of misassemblies detected for each value of the threshold T is the vertical difference between the experimental (blue) curve and the fit (magenta). Figure 1 implies that there are about *9* expansion misassemblies for the $C>4.7$ and about *385* compression misassemblies for $C<-4$.

Assembly comparison/reconciliation.

We developed a preliminary version of software that takes two assemblies, and creates a composite assembly, which has fewer gaps and misassemblies than either one of the initial assemblies. It patches gaps and CE points using pieces of another assembly. This is an initial step. There are many more possible ways of reconciling pairs of assemblies that it is not yet designed to perform. This software is currently capable of handling genomes up to 250Mb, which includes genomes of fly species. The process can be applied recursively using several different assemblies to enhance the original draft assembly. In what follows we list the concepts, which the software is based on and our preliminary results.

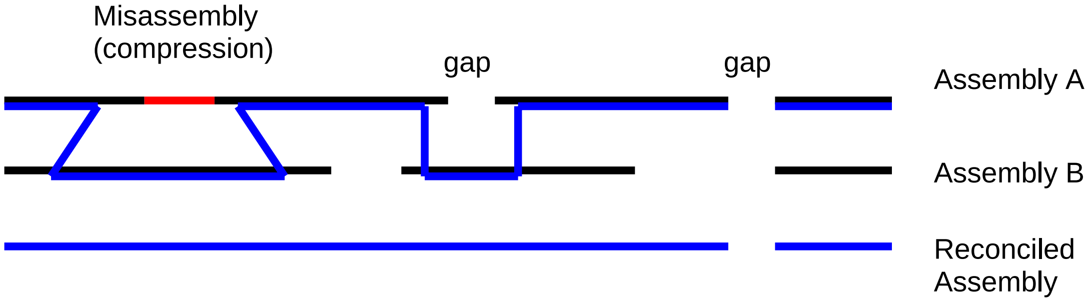


Figure 2. Illustration of assembly reconciliation concept. The blue line shows the reconciled assembly; it avoids a misassembly in Assembly A and a gap in Assembly B.

Figure 2 shows the strategy for assembly reconciliation. The reconciliation is based on 2 methods: detecting misassemblies and closing gaps. We currently use the CE statistic to detect compression/expansion misassemblies and to verify the validity of gap spanning.

The algorithm that we currently use to reconcile 2 assemblies (we call them the *reference assembly* and the *supplementary assembly*) is as follows:

1. Compute the CE statistic on the reference assembly and break it at the CE problem points, introducing positive gaps for the compressions and negative gaps for expansions, creating a gapped reference assembly.
2. Align gapped reference assembly to the supplementary assembly using NUCmer (Delcher et al. 1999, Delcher et al. 2002, Kurtz 2004).
3. Find out which contigs in the supplementary assembly span the gaps within scaffolds of the reference assembly (both pre-existing gaps and gaps introduced

in step 1) such that (i) the gap size with respect to the alignment is within 3 reported standard deviations of the reported scaffold gap size, and (ii) the supplementary assembly has good CE score in the span region.

4. Attempt to close the gaps that were spanned by the supplementary assembly by finding common read sequences to go into the gaps. Record which gaps were closed successfully. Record which gap closures resulted in introducing new CE problem points.
5. Go back to the reference assembly and only introduce the CE gaps that were successfully closed by in step 4.
6. Close only those gaps that did not introduce new CE problems and patch the consensus sequence based on the alignment coordinates.
7. Adjust the positioning of reads so that no inserted read occurs more than once in the assembly.

We have applied the software to the Aug. 2005 *D. virilis* assembly by Agencourt that was created with the Arachne assembler (reference) and the VI assembly of August 2005 that was created by the Venter Institute (supplementary). We then further enhanced the resulting reconciled assembly with an assembly that was made using a very conservative set of UMD overlaps from UMD Overlapper with Celera Assembler. We measured two parameters: the number of CE problem points and N50 contig size (N50 is the contig size such that the contigs larger than that have 50% the bases of the assembly). Notice that closing a pre-existing gap merges two contigs and increases the average size of the contigs and generally increases N50. The before and after statistics are as follows:

The initial data on the Agencourt Aug 2005 assembly:

1. CE Problem locations: 1566
2. N50 contig size: 101Kb

After reconciliation with VI assembly:

1. CE Problem locations: 1245
Fixed problem locations: 321, or 20%
2. N50 contig size: 115Kb

After additional reconciliation with an assembly created by Celera Assembler with UMD overlaps using the same initial data:

1. CE Problem locations: 1078
Fixed problem locations: 488, or 31%
2. N50 contig size: 118Kb

The results above suggest that assemblies can be improved significantly using assembly reconciliation. In the software that we have developed so far we have only taken a first few steps: closing gaps in scaffolds and fixing some CE problem points.